Project Plan

CyberQuest: Interactive Platform for Teaching Cybersecurity

Team Members: Matthew Goembel, Anthony Clayton, Ludendorf Brice, Ben Allerton

Faculty Advisor: Sneha Sudhakaran, ssudhakaran@fit.edu

Client: Sneha Sudhakaran, College of Engineering and Science

Date(s) of Meeting(s) with the Client for Developing this Plan

Project Goals

Provide an interactive and effective platform to teach cybersecurity to users of all age groups (children, teenagers, adults, and professionals). Build user awareness, practical skills, and resilience against cyber threats. Allow compatibility with all the most commonly used operating systems and languages. Offer hands-on experience with simulations of real-world scenarios to build real-world technical skills.

Project Motivation

Lack of Accessibility and Usability: Existing cybersecurity learning platforms often use overly technical language, intimidating interfaces, or require prior knowledge, discouraging especially younger and inexperienced audiences, from engaging in cybersecurity education.

Absence of Gamification and Fun: Cybersecurity is often presented in a dry, textbook-like manner. Without gamification and interactivity, users struggle to stay motivated, and learning outcomes are limited.

Limited Age-Appropriate Content: Current platforms don't cater to diverse age groups effectively. A common one-size-fits-all approach leaves children overwhelmed, teenagers disengaged, and adults feeling the content is either oversimplified or overly technical.

Fear and Intimidation Around Cybersecurity Concepts: Many users find complex cybersecurity concepts abstract and intimidating. Instead of avoiding the topic entirely, a guided approach can help users feel confident and capable.

Lack of hands-on opportunities: Without practical exposure, users cannot effectively apply cybersecurity principles in real-life scenarios.

Lack of Global Reach and Multilingual Support: Most existing platforms focus on English-speaking audiences. A multilingual, culturally adaptable approach would expand the reach of cybersecurity education globally.

Approach (key features of the system)

Interactive tutorials and quizzes: Users can engage with interactive tutorials that guide them through essential cybersecurity concepts in a step-by-step manner. The users can then apply their knowledge through quizzes at the end of each tutorial, which tests their understanding using multiple-choice, drag-and-drop, and scenario-based questions. This feature helps users actively participate in the learning process and strengthens their ability to retain the material.

Age-appropriate modules with gamification: The users can access age-appropriate modules with games tailored to their specific learning level, from fun, introductory content for children to more complex lessons for professionals. Users can interact with gamified elements, such as playful characters for younger learners or simulated cybersecurity challenges for adults and teens. This gamification enhances the user experience and keeps users engaged while helping them master cybersecurity skills.

Real-world cybersecurity challenges: Users can learn by playing in real-world cybersecurity simulations, where they simulate defending against cyberattacks like phishing, malware, and social engineering. The users can practice applying cybersecurity strategies in scenarios that mirror real-life situations, helping them develop practical, hands-on skills. Additionally, users can potentially analyze case studies of cyber incidents, gaining insights into how cybersecurity threats evolve and how to protect against them.

Progress tracking and certifications: The users can track their learning progress by monitoring their completion of tutorials and quizzes. Users can earn badges, achievements, and certifications as they progress through different levels of content, completing specific milestones or advanced courses, providing them with a tangible way to measure their learning success and motivation to continue improving their cybersecurity knowledge.

Support for multiple languages: Users can access the platform in multiple languages, ensuring that content is available in their preferred language. The users can interact with all tutorials, quizzes, and challenges in a way that is culturally and linguistically relevant to them, making cybersecurity education more accessible to a global audience, and ensuring that language and region barriers are not a hindrance to learning.

Novel Features/Functionalities

Multilingual Support: While many cybersecurity platforms exist, not all of them cater to global audiences by offering multilingual support. By offering support for multiple languages, you're making cybersecurity education more inclusive and accessible on a global scale.

Real-World Interactive Simulations: While some platforms provide theoretical content, interactive, hands-on simulations where users can defend against phishing attacks, malware, and other threats mirror real-life cybersecurity challenges. Helps learners apply theoretical knowledge in real-world scenarios, building practical experience that is crucial in cybersecurity but often missing in other platforms.

Progress Tracking and Certifications: Many platforms offer courses, but progress tracking and certifications that reflect user achievements in a gamified and interactive environment are less common. Tracking progress across different skill levels and offering certificates upon completion creates motivation and gives users something tangible to show for their efforts.

Adaptive Learning Paths: Based on a learner's progress and performance, the platform could automatically adjust difficulty levels or give personalized learning suggestions, ensuring the content is always appropriately challenging.

Algorithms and Tools

Game Development: Unity (GDScript, C#), Phaser (JavaScript), Docker, GamifyJS.

Front-End: HTML, CSS, JavaScript, React.js.

Back-End: Python, Node.js, Java.

Database & Storage: PostgreSQL, MySQL, MongoDB.

Hosting & Deployment: GitHub Pages, AWS, Azure, ExpressJS, Render.

Testing & Debugging: Selenium for automated testing, unit, and functional testing.

Authentication: OAuth 2.0, JWT (JSON Web Tokens), bcrypt for secure password hashing.

Collaboration & Organization: GitHub (version control), Jira (task management), Discord, iMessage, email (communication).

Technical Challenges

Defining the Structure of Real-World Cybersecurity Simulations: Designing realistic yet engaging scenarios for threats like phishing and malware is challenging, especially in balancing complexity for diverse audiences. We must identify the best tools and frameworks and the methods for maintaining simplicity and efficiency.

Implementing Multilingual Support and Localization: Establishing a reliable system for accurate translations of cybersecurity terminology is challenging. Deciding on the tools and libraries to use for language handling, and handling updates/modifications to translated content without disrupting user experience serves as a hurdle.

Developing Frontend and Backend Architecture: Selecting technologies like React.js and Python, designing efficient APIs for progress tracking and gamified features, and addressing performance issues for real-time simulations are more pressing challenges.

Implementing Secure Authentication and Authorization: Designing a secure and user-friendly authentication system that handles multiple methods, such as email/password login and third-party login providers (e.g., Google or Microsoft).

Designing Effective Cybersecurity Education Content: Developing interactive, engaging, and age-appropriate lessons involves technical challenges such as integrating dynamic educational content into gamified systems and ensuring compatibility with various devices and operating systems. Adapting content to real-world scenarios while maintaining a balance between technical depth and accessibility requires advanced algorithms for adaptive learning paths. Additionally, implementing tools to track user comprehension and progression across diverse topics poses technical hurdles.

Milestone 1 (Feb 24): Initial Setup and Technical Evaluation

Itemized Tasks:

1. **Compare and Select Technical Tools**:
    - Evaluate front-end frameworks (React.js, Angular, etc.).
    - Compare back-end frameworks (Node.js, expressJS, etc.).
    - Decide on database options (PostgreSQL, MongoDB, etc.).
    - Authentication: Evaluate OAuth 2.0, JWT (JSON Web Tokens), bcrypt.
    - Assess gamification tools (Unity, Phaser, GamifyJS).
2. **"Hello World" Demos**:
    - Frontend: Create a functional webpage.
    - Backend: Set up a simple server and database with a test API endpoint.
    - Gamification: Develop a basic working game in Phaser or Unity.
    - Authentication: Demonstrate basic login functionality.
3. **Resolve Technical Challenges**:
    - Finalize frameworks for real-world simulations and gamification.

    - Choose localization tools to implement ADA-compliant, multilingual support.
    - Identify tools for seamless integration between the frontend, backend, and games.
    - Address integration of authentication tools with the front end and back end.
4. **Select Collaboration Tools**:
    - Compare and finalize tools like GitHub, Jira, and Discord for task management and communication.
5. **Draft Initial Documents**:
    - Requirement Document: Define core features, technical specifications, and target users.
    - Design Document: Outline system architecture and workflows.
    - Test Plan: Develop a preliminary testing strategy for unit, integration, and usability testing.

Milestone 2 (Mar 26): Feature Implementation and Testing

Itemized Tasks:

1. **Implement, Test, and Demo Core Features**:
    - Frontend: Develop interactive tutorials and quizzes for one user group (e.g., children).
    - Backend: Implement APIs for user authentication, progress tracking, and quiz scoring.
    - Gamification: Integrate age-appropriate games for one user group into the front end.
    - Authentication system: User login, registration, and secure password handling. Integrate with Google (e.g. login using a Google account).
2. **Address Localization and Accessibility**:
    - Add multi-language support for at least two languages.
    - Ensure the platform complies with ADA accessibility standards.
3. **Progress Tracking**:
    - Develop a progress-tracking system that connects the backend to the front end.
    - Test user accounts for data persistence and security.
4. **Validate System Performance**:
    - Perform load testing on APIs and game components.
    - Debug and resolve latency issues in real-time simulations.
5. **Documentation**:
    - Update Requirements and Design Documents with finalized features.
    - Refine the Test Plan based on new features and performance data.

Milestone 3 (Apr 21): Final Implementation and Launch Prep

Itemized Tasks:

1. **Implement, Test, and Demo Advanced Features**:
    - Add remaining user groups (teenagers, adults, professionals) with tailored tutorials and games.
    - Complete real-world cybersecurity simulations (phishing, malware defense, etc.).
    - Finalize certifications and badge system.
    - Refine authentication: Ensure robust security and user account management.
2. **Integrate Multilingual**:
    - Extend language support to more languages.
3. **Deployment Preparation**:
    - Configure hosting platform (AWS, Azure, or GitHub Pages, ExpressJS, Render).
    - Set up HTTPS for secure connections.
4. **Testing and QA**:
    - Conduct end-to-end testing for all modules.
    - Perform user testing to gather feedback on usability and accessibility.
5. **Finalize Documentation**:
    - Submit complete Requirements, Design, and Test Documents.
    - Prepare user manuals and deployment guides for the platform.
6. **Prepare for Client Presentation**:
    - Develop a walkthrough demo showcasing key features.
    - Collect usage data and metrics for the client report.

Task Matrix for Milestone 1

Task	**Anthony**	**Matthew**	**Ludendorf**	**Ben**
Compare and select Technical Tools	Backend, Database	Gamification	Authentication	Frontend
"hello world" demos	Set up a simple server and database with a test API endpoint	Develop a basic working game in Phaser or Unity	Demonstrate basic login functionality	Create a functional webpage
Resolve Technical Challenges	Developing Database and Backend Architecture	Defining the Structure of Real-World Cybersecurity Simulations	Implementing Secure Authentication and Authorization, Designing Effective Cybersecurity Education Content	Implementing Multilingual Support and Localization, Develop Frontend Architecture

Compare/select Collaboration Tools	Task calendar, programs	documentation, presentations, communication		
Requirement Document	Write 20%	Write 50%	Write 20%	Write 10%
Design Document	Write 40%	Write 20%	Write 20%	Write 20%
Test Plan	Write 30%	Write 20%	Write 20%	Write 30%

Approval from Faculty Advisor

"I have discussed with the team and approve this project plan. I will evaluate the progress and assign a grade for each of the three milestones."

Signature: _____ Date: _____